REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
28-06-2011	FINAL	01-04-2008 - 31-03-2011

4. TITLE AND SUBTITLE

Adaptive Optimization Techniques for Large-Scale Stochastic Planning

5a. CONTRACT NUMBER

FA9550-08-1-0171

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Zilberstein, Shlomo

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Massachusetts
Department of Computer Science
140 Governors Drive
Amherst, MA 01003-9264

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Grant and Contract Administration
70 Butterfield Terrace
University of Massachusetts
Amherst, MA 01003

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-OSR-VA-TR-2012-0248

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author, and not necessarily shared by AFOSR.

14. ABSTRACT

Solving large Markov Decision Problems (MDPs) is a very useful, but computationally challenging problem addressed widely in reinforcement learning and operations research. The commonly used approximation methods can be divided into three categories: policy search, approximate dynamic programming, and approximate linear programming. We developed a new approximate bilinear programming (ABP) formulation of value function approximation, which employs global optimization. The formulation provides strong a priori guarantees on both robust and expected policy loss by minimizing specific norms of the Bellman residual. Solving a bilinear program optimally is NP-hard, but this is unavoidable because the Bellman-residual minimization itself is NP-hard. We constructed and analyzed both optimal and approximate algorithms for solving bilinear programs. These algorithms offer a convergent generalization of approximate policy iteration. In practice, only sampled versions of ABPs are often solved. We examined the impact of sampling and established worst-case error bounds. In experimental work, we demonstrated that the new approach could consistently minimize the Bellman residual on several benchmark problems.

15. SUBJECT TERMS

Markov decision processes; planning under uncertainty; approximate bilinear programming; approximate dynamic programming; reinforcement learning; stochastic optimization

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Shlomo Zilberstein
U	U	U	SAR	8	19b. TELEPHONE NUMBER *(Include area code)* 413-545-4189

Reset

Final Performance Report

Adaptive Optimization Techniques for Large-Scale Stochastic Planning

AFOSR Agreement Number FA9550-08-1-0171

Shlomo Zilberstein, Principal Investigator
Computer Science Department
140 Governors Drive
University of Massachusetts
Amherst, MA 01003-9264

June 28, 2011

1 Overview

We developed new optimization-based methods for stochastic planning that offer better performance and better convergence guarantees compared to the state-of-the-art AI methods. In AI, reinforcement learning algorithms have proved useful in many complex domains, such as resource management and planning under uncertainty. These algorithms are often iterative—they successively approximate the solution based on a set of samples and features. Although these iterative algorithms can achieve impressive results in some domains, they have substantial drawbacks: they often require extensive parameter tweaking to work well and provide only weak guarantees of solution quality. Some of the most interesting reinforcement learning algorithms are based on approximate dynamic programming (ADP). ADP, also known as value function approximation, approximates the value of being in each state. This project produced new reliable algorithms for ADP that use optimization instead of iterative improvement. Because these optimization-based algorithms explicitly seek solutions with favorable properties, they are easy to analyze, offer much stronger guarantees than iterative algorithms, and have few or no parameters to tweak. In particular, we derive approximate bilinear programming—a new robust approximate method. The strong guarantees of optimization-based algorithms not only increase confidence in the solution quality, but also make it easier to combine the algorithms with other ADP components, most notably samples and features used to approximate the value function. Relying on the simplified analysis of optimization-based methods, we derived new bounds on the error due to missing samples. These bounds are simpler, tighter, and more practical than the existing bounds for iterative algorithms and can be used to evaluate solution quality in practical settings. Finally, we developed homotopy methods that use the sampling bounds to automatically select good approximation features for optimization-based algorithms. Automatic feature selection significantly increases the flexibility and applicability of the developed ADP methods. The methods developed in this project can be used in many practical applications in artificial intelligence, operations research, and engineering. Our experimental results show that optimization-based methods may perform well on resource-management problems and standard benchmark problems and therefore represent an attractive alternative to traditional iterative methods.

2 Summary of Research Challenges and Accomplishments

Automatic planning in large domains is one of the hallmarks of intelligence and a core research area of artificial intelligence. Being able to adaptively plan in an uncertain environment can significantly reduce costs, improve efficiency, and relieve human operators from many mundane tasks. We targeted a range of applications represented by the following two domains that illustrate the utility of automated planning and the challenges involved.

One application is management of blood inventories. In this domain, a blood bank aggregates a supply of blood and keeps an inventory to satisfy hospital demands. The hospital demands are stochastic and hard to predict precisely. In addition, blood ages when it is stored and cannot be kept longer than a few weeks. The decision maker must decide on blood-type substitutions that minimize the chance of future shortage. Because there is no precise model of blood demand, the solution must be based on historical data. Even with the available historical data, calculating the optimal blood-type substitution is a large stochastic problem.

Another application is managing water reservoirs. In this domain, an operator needs to decide how much and when to discharge water from a river dam in order to maximize energy production, while satisfying irrigation and flood control requirements. The challenges in this domain are in some sense complementary to blood inventory management with fewer decision options but greater uncertainty in weather and energy prices.

Many practical planning problems such as the ones mentioned above are solved using domain-specific methods. This entails building specialized models, analyzing their properties, and developing specialized algorithms. For example, blood inventory and reservoir management could be solved using the standard theory of inventory management. The drawback of specialized methods is their limited applicability. Applying them requires significant human effort and specialized domain knowledge. In addition, the domain can often change during the lifetime of the planning system. Domain-specific methods may also be inapplicable if the domain does not clearly fit into an existing category. For example, because of the compatibility constraints among blood types, blood inventory management does not fit well the standard inventory control framework. In reservoir management, the domain-specific methods also do not treat uncertainty satisfactorily, nor do they work easily with historical data. This project produced *general* planning method that are easy to apply to a variety of settings as an alternative to domain-specific ones. Having general methods that can reliably solve a large variety of problems in many domains would enable widespread application of planning techniques.

In this project, we used Markov decision process (MDP) to represent complex sequential decision making problems. Although MDPs are easy to formulate, they are often very hard to solve. Solving large MDPs is a computationally challenging problem addressed widely in artificial intelligence (particularly reinforcement learning), operations research, and engineering literature. It is widely accepted that large MDPs can only be solved approximately. Approximate solutions may be based on samples of the domain, rather than the full descriptions.

An MDP consists of a set of states, S, and a set of actions, A. After each action is taken, a stochastic state transition occurs and a reward is given to the decision maker that depends on the action and outcome. The solution of an MDP is a policy ßthat assign an action to each state. A related solution concept is the value function, v, which represents the expected value of being in every state. A value function can be easily used to construct a *greedy* policy. It is useful to study value functions, because they are easier to analyze than policies. For any policyß, the policy loss is the difference between the return of the policy and the return of an optimal policy. Because it is often not feasible to compute an optimal policy, the goal of this project has beento compute a policy with a small policy loss.

Approximate methods for solving MDPs can be divided into two broad categories: 1) policy search, which explores a restricted space of all policies, 2) approximate dynamic programming, which searches a

restricted space of value functions. While all of these methods have achieved impressive results in many domains, they have significant limitations that we address is this project.

Policy search methods rely on local search in a restricted policy space. The policy may be represented, for example, as a finite-state controller or as a greedy policy with respect to an approximate value function. Policy search methods have achieved impressive results in such domains as Tetris and helicopter control. However, they are notoriously hard to analyze. We are not aware of any theoretical guarantees regarding the quality of the solution.

Approximate dynamic programming (ADP)—also known as value function approximation—is based on computing value functions as an intermediate step before computing polices. Most ADP methods iteratively approximate the value function. Traditionally, ADP methods are defined procedurally; they are based on precise methods for solving MDPs with an approximation added. For example, approximate policy iteration—an approximate dynamic programming method—is a variant of policy iteration. The procedural approach leads to simple algorithms that may often perform well. However, these algorithms have several theoretical problems that make them impractical in many settings.

Although procedural (or iterative) ADP methods have been extensively studied and analyzed, their is still limited understanding of their properties. They do not converge and therefore do not provide finite-time guarantees on the size of the policy loss. As a result, procedural ADP methods typically require significant domain knowledge to work; for example they are sensitive to the approximation features. The methods are also sensitive to the distribution of the samples used to calculate the solution and many other problem parameters. Because the sensitivity is hard to quantify, applying the existing methods in unknown domains can be very challenging.

This project produced a new optimization-based approach to approximate dynamic programming as an alternative to traditional iterative methods. Unlike procedural ADP methods, optimization-based ADP methods are defined declaratively. In the declarative approach to ADP, we first explicitly state the desirable solution properties and then develop algorithms that can compute such solution. This leads to somewhat more involved algorithms, but ones that are much easier to analyze. Because these optimization techniques are defined in terms of specific properties of value functions, their results are easy to analyze and they provide strong guarantees. In addition, the formulations essentially decouple the actual algorithm used from the objective, which increases the flexibility of the framework.

The objective of optimization-based ADP is to compute a value function v that leads to a policy with a small policy loss. Unfortunately, the policy loss as a function of the value function lacks structure and cannot be effiefficiently computed without simulation. We, therefore, derived upper bounds on the policy loss that are easy to evaluate and optimize. Approximate linear programming (ALP), which can be classified as an optimization-based approach to ADP, has been proposed and studied previously. ALP uses a linear program to compute the approximate value function in a particular vector space. ALP has been previously used in a wide variety of settings and has shown to have better theoretical properties than iterative approximate dynamic programming and policy search. However, the L_1 norm must be properly weighted to guarantee a small policy loss, and there is no reliable method for selecting appropriate weights. Among other contributions, this project produced modifications of ALP that improve its performance and methods that simultaneously optimize the weights with the value function.

Value function approximation—or approximate dynamic programming—is only one of many components that are needed to solve large MDPs. Other important components are the domain samples and features, or representation, used to approximate the value function. The features represent the prior knowledge. It is desirable to develop methods that are less sensitive to the choice of the features or are able to discover them automatically. It is easier to specify good features for optimization-based algorithms than for iterative value function optimization. The guarantees on the solution quality of optimization-based methods can be used to guide feature selection for given domain samples.

The main contribution of this project is the formulation and study of optimization-based methods for

approximate dynamic programming. The project also investigated how these methods can be used for representation and feature selection. The contributions are summarized below:

1. New and improved optimization-based methods for approximate dynamic programming.
   (a) New bounds on the quality of an approximate value function.
   (b) Lower bounds on the performance of iterative approximate dynamic programming.
   (c) Improved formulations of approximate linear programs.
   (d) Tight bilinear formulation of value function approximation.
2. Algorithms for solving optimization-based dynamic programs.
   (a) Homotopy continuation methods for solving optimization-based formulation.
   (b) Approximate algorithms for optimization-based formulations.
   (c) Methods for more efficiently solving some classes of bilinear programs involved in value function approximation.
3. Methods for selecting representation.
   (a) Sampling bounds for optimization-based methods.
   (b) Representation selection based on sampling bounds and the homotopy methods.
4. Connections between value function approximation and classical planning.

These outcomes are described in details in archival publications listed in the following section.

3 Publications

Note: The publications are available for download at:
`http://rbr.cs.umass.edu/shlomo/`

3.1 PhD Dissertations

1. Marek Petrik. "Optimization-based Approximate Dynamic Programming." PhD Dissertation, Computer Science Department, University of Massachusetts Amherst, 2010.

3.2 Journals and Conferences

1. M. Petrik and S. Zilberstein. "A Successive Approximation Algorithm for Coordination Problems." *Proceedings of the Tenth International Symposium on Artificial Intelligence and Mathematics (ISAIM-08)*, Ft. Lauderdale, Florida, 2008.
2. M. Allen, M. Petrik, and S. Zilberstein. "Interaction Structure and Dimensionality in Decentralized Problem Solving." *Proc. of the Conference on Artificial Intelligence*, 2008.
3. M. Allen, M. Petrik, and S. Zilberstein. "Interaction Structure and Dimensionality in Decentralized Problem Solving." *proc. of the Conference on Artificial Intelligence*, 2008. Extended version of AAAI-08 paper published as Technical Report UM-CS-2008-011, University of Massachusetts, Amherst, 2008.

4. M. Petrik and S. Zilberstein. "Learning Heuristic Functions Through Approximate Linear Programming." *Proc. of the International Conference on Automated Planning and Scheduling* (ICAPS), 248–255, 2008.

5. M. Petrik and B. Scherrer. "Biasing Approximate Dynamic Programming with a Lower Discount Factor." *Proc. of Neural Information Processing Systems*, 2008.

6. M. Petrik and S. Zilberstein. "A Bilinear Programming Approach for Multiagent Planning." *Journal of Artificial Intelligence Research*, 35:235–274, 2009.

7. M. Petrik and S. Zilberstein. "Constraint Relaxation in Approximate Linear Programs." *Proc. of the International Conference on Machine Learning* (ICML), 809–816, 2009.

8. M. Petrik and S. Zilberstein. "Robust Value Function Approximation Using Bilinear Programming." *Proc. of Neural Information Processing Systems* (NIPS), 2009.

9. M. Petrik and S. Zilberstein. "Robust Value Function Approximation Using Bilinear Programming." Extended version of NIPS-09 paper published as Technical Report UM-CS-2009-052, University of Massachusetts, Amherst, 2009.

10. M. Petrik. "Robust Approximate Optimization for Large Scale Planning Problems." AAAI Doctoral Consortium, 2009.

11. J. Johns, M. Petrik, and S. Mahadevan. "Hybrid Least-Squares Algorithms for Approximate Policy Evaluation." *Machine Learning*, 76(2-3): 243–256, 2009.

12. M. Petrik and S. Zilberstein. "Blood Management Using Approximate Linear Programming." Presented at INFORMS Computing Society Meeting, Charleston, SC, 2009.

13. M. Petrik, G. Taylor, R. Parr, and S. Zilberstein. "Feature Selection Using Regularization in Approximate Linear Programs for Markov Decision Processes." *Prof. of the International Conference on machine Learning* (ICML), 871–878, 2010.

14. M. Petrik and S. Zilberstein. "Robust Approximate Bilinear Programming for Value Function Approximation." Submitted to *Journal of Machine Learning Research*, 2010.

15. M. Petrik and S. Zilberstein. "Linear Dynamic Programs for Resource Management." *Proc. of the Conference on Artificial Intelligence* (AAAI), 2011

4 Interactions and Transitions

The project team was very active in several conferences, symposia, panels, and journals. The main graduate student assigned to this project, Marek Petrik, has completed his PhD dissertation and is now a postdoc at IBM Research. Team members were engaged in several international collaborations. These interactions, which help disseminate the results of the project, are summarized below.

4.1 Editorial Positions

1. The PI is currently the Editor-in-Chief of the *Journal of Artificial Intelligence Research*, one of the top journals in the field of AI. He has been serving on the editorial board of the journal since 2002.

2. The PI serves on the editorial board of two other journals: *Autonomous Agents and Multi-Agent Systems* and *Annals of Mathematics and Artificial Intelligence*.

4.2 Participation in Conference and Workshop Organization

The PI and members of this project team served extensively on the program committees of the following venues.

1. Twenty-Fifth AAAI Conference on Artificial Intelligence, July 7-11, 2011, San Francisco, California.
2. AAAI 2011 Workshop on Generalized Planning, August 8, 2011, San Francisco, California.
3. Twenty-Second International Joint Conference on Artificial Intelligence, July 16-22, 2011, Barcelona, Spain.
4. IJCAI 2011 Workshop on Decision Making in Partially Observable, Uncertain Worlds: Exploring Insights from Multiple Communities, July 18, 2011, Barcelona, Spain.
5. Twenty-First International Conference on Automated Planning and Scheduling, June 11-16, 2011, Freiburg, Germany.
6. Tenth International Conference on Autonomous Agents and Multiagent Systems May 2-6, 2011, Taipei, Taiwan.
7. Twenty-Fourth AAAI Conference on Artificial Intelligence, July 11-15, 2010, Atlanta, Georgia.
8. AAAI 2010 Workshop on Metacognition for Robust Social Systems, July 11-12, 2010, Atlanta, Georgia.
9. Second International Conference on Computational Sustainability, June 28-30, 2010, Boston, Massachusetts.
10. Twentieth International Conference on Automated Planning and Scheduling, May 12-16, 2010, Toronto, Canada.
11. AAMAS 2010 Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains, May, 2010, Toronto, Canada.
12. Eleventh International Symposium on Artificial Intelligence and Mathematics, January 6-8, 2010, Fort Lauderdale, Florida.
13. International Conference on Automated Planning and Scheduling, September 19-23, 2009, Thessaloniki, Greece.
14. ICAPS 2009 Workshop on Generalized Planning: Macros, Loops, Domain Control, September 20, 2009, Thessaloniki, Greece.
15. Twenty-First International Joint Conference on Artificial Intelligence, July 11-17, 2009, Pasadena, California.
16. Eighth International Conference on Autonomous Agents and Multiagent Systems, May 10-15, 2009, Budapest, Hungary.
17. AAMAS 2009 Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains, May 11, 2009, Budapest, Hungary.
18. International Conference on Automated Planning and Scheduling, September 14-18, 2008, Sydney, Australia.
19. ICAPS 2008 Workshop on Multiagent Planning, September 14 or 15, 2008, Sydney.
20. Twenty-Third AAAI Conference on Artificial Intelligence, July 13-17, 2008, Chicago, Illinois.
21. The First International Symposium on Search in Artificial Intelligence and Robotics, July 13-14, 2008, Chicago, Illinois.
22. AAAI 2008 Workshop on Metareasoning: Thinking about Thinking, July 13-14, 2008, Chicago, Illinois.

23. Seventh International Conference on Autonomous Agents and Multiagent Systems, May 12-16, 2008, Estoril, Portugal.
24. AAMAS 2008 Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains, May 12-13, 2008, Estoril, Portugal.

4.3 Other Interactions

1. The PI has maintained close collaboration ties between his lab and the MAIA group at INRIA, Nancy, France. To advance this collaboration, INRIA has provided funding for exchange of students and short visits. The PI has also participated in a multi-institutional NSF grant that provided additional funding for this collaboration. These activities contributed directly to this project, particular the joint work between our group and Bruno Scherrer from INRIA.

2. The PI is currently the president of the ICAPS Executive Council, which oversees the annual International Conference on Automated Planning and Scheduling—the premier venue for researchers and practitioners in the area of planning and scheduling.

5 Inventions and Patent Disclosures

None.

6 Honors and Awards

1. The PI was elected as a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI) for "significant contributions to decision-theoretic reasoning, resource-bounded reasoning, automated planning, decentralized decision making and multi-agent systems." The AAAI will celebrate this honor at a Fellows dinner during AAAI-11 in San Francisco, California.